Underfitting and Overfitting

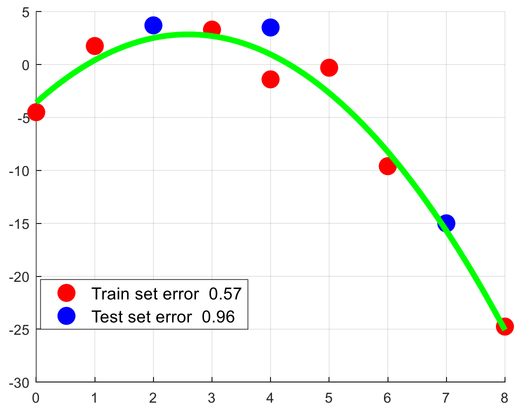
Underfitting - the prediction model selected by the algorithm is too simplistic to represent the relationship in the dataset between the descriptive features and the target feature.

Overfitting - the prediction model selected by the algorithm is so complex that the model fits to the dataset too closely and becomes sensitive to noise in the data

Example

Underfitting

Just right



Overfitting

Methods to prevent overfitting

- Cross validation
- More data (data augmentation, …)
- Remove irrelevant features
- Early stopping
- Regularization (weight decay, …)
- Ensemble (collective decision)

Evaluation of the model (scope)

DATA

Training set | Test set

DECISION MAKING MODEL

Results

Different data splitting techniques are used based on amount of available data

Performance Measures

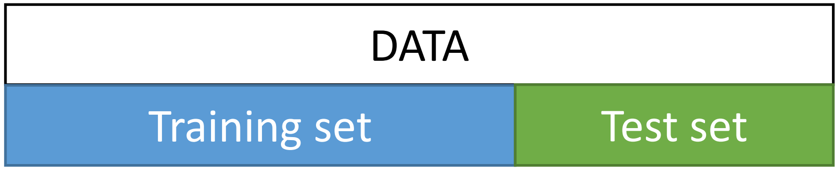
Different performance measurement techniques for different decision problems: classification, regression, object detection, image segmentation, …

EVALUATION

Evaluate model with data, that has ***not been used*** *to train the model!*

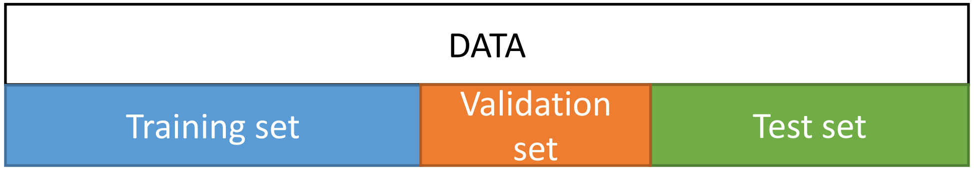
Holdout Method

Remove a part of the training data and use it to get predictions from the model trained on rest of the data.



Splits 70%-30% / 60%-40% / …

Validation set is used if data outside of the training set is required in order to tune particular aspect of the model, i. e. define combination of the most appropriate hyper parameters for the problem under consideration.



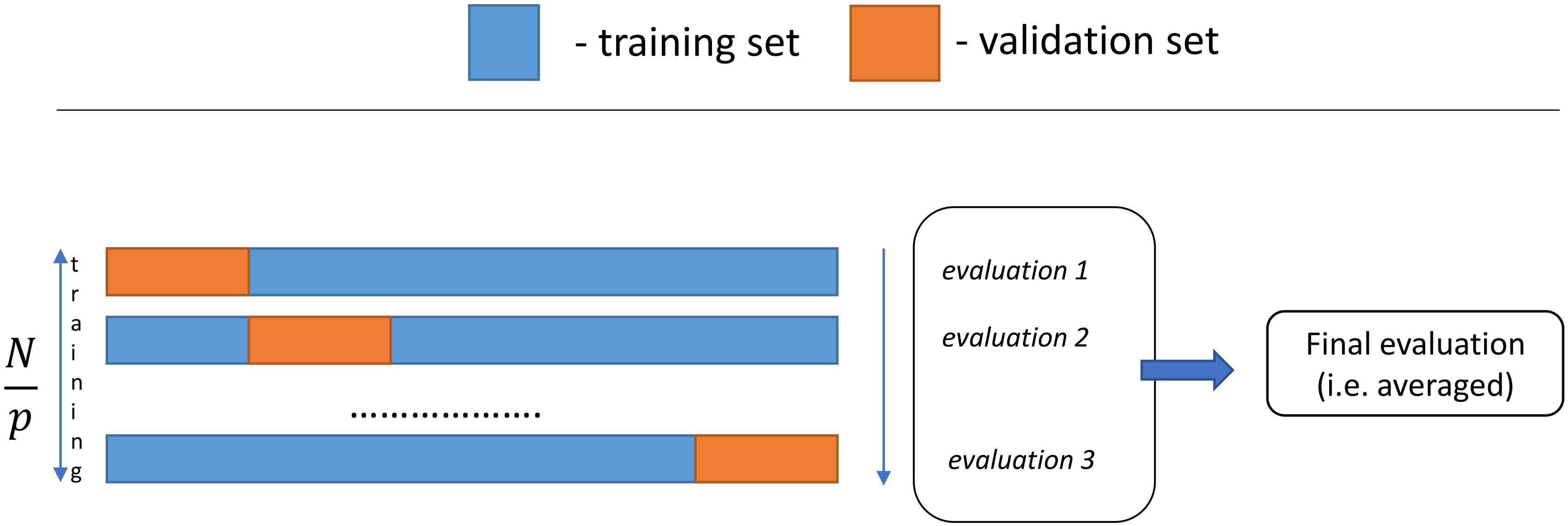
Splits 70%-30% / 60%-40% / …

Other sampling techniques

Leave p-out

Exhaustive cross-validation

For the dataset of m samples, p are left out as validation set and m-p are used for training. Special case – leave-1-out

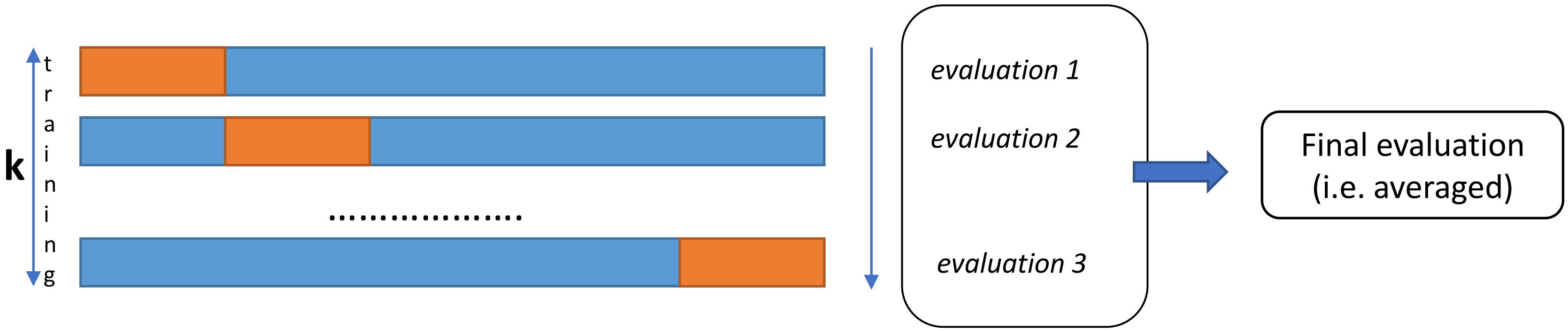


Other sampling techniques

k-Fold cross-validation
Non-exhaustive cross-validation

dataset is shuffled and randomly splitted into k equal-sized folds (partitions), and k separate evaluation experiments are performed

- training set - validation set

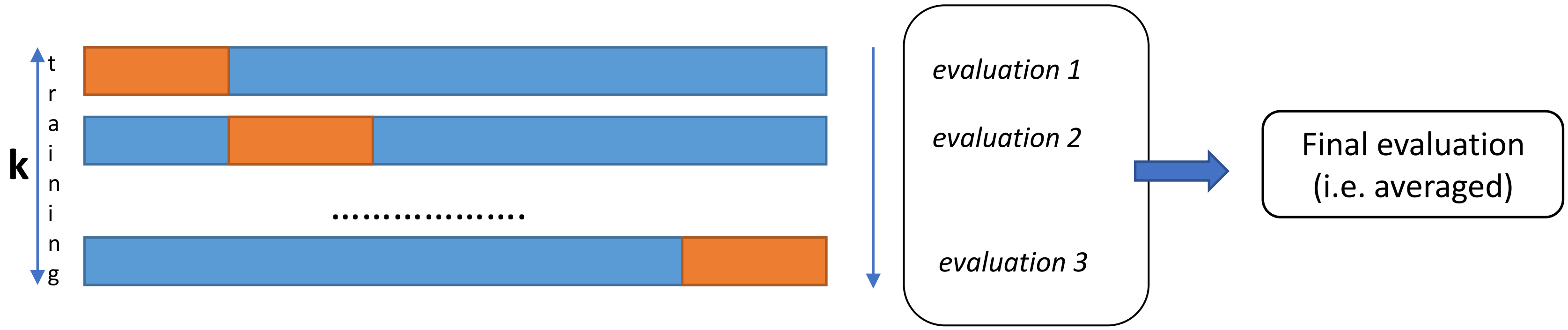


Other sampling techniques

Stratified k-Fold cross-validation
Non-exhaustive cross-validation

A variation of k-Fold cross validation such that each fold contains similar percentage of each class (or approximately equal mean values for regression problems) as the complete dataset.

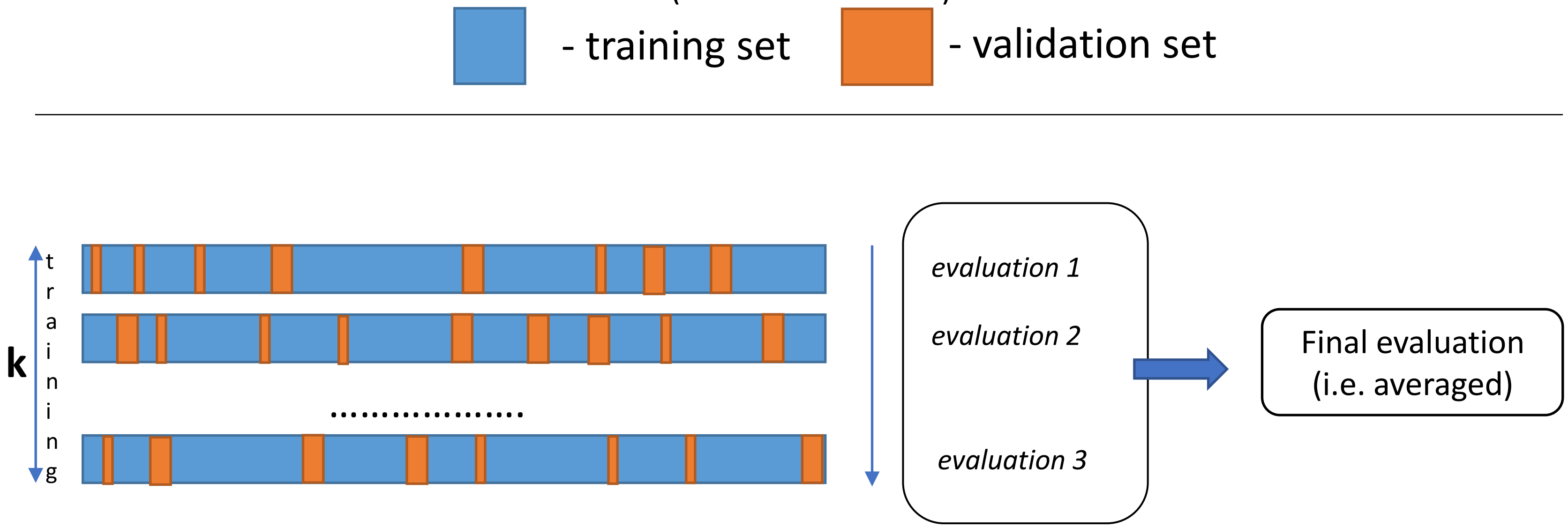
- training set - validation set



Other sampling techniques

Bootstrapping
Non-exhaustive cross-validation

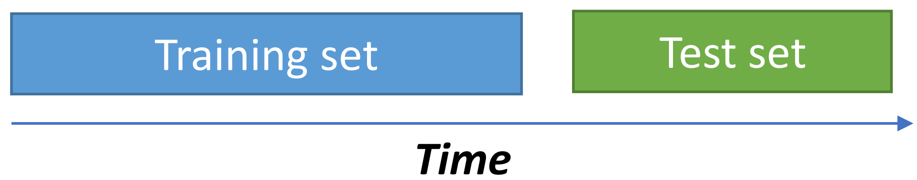
Iteratively perform multiple evaluation experiments using slightly different training and test sets (preferred for very small datasets). Typically k is set much larger than k in k-fold cross validation (i.e. more than 200)



Other sampling techniques

*In scenarios that include a time dimension **out-of-time** sampling **commonly is used,** when splitting includes a time dimension.*

Out-of-time sampling



Such splitting applicable i.e. if customers‘s behavior is forecasted for forthcoming year based on past one-year period

Metrics

Decision Model Performance Measurement (Classification)

Confusion matrix is a table, which is often used to describe the performance of a classification model on a set of test data for which the true values are known.

- **True Positive (TP)** - a <u>positive</u> target feature value and <u>correct</u> prediction
- **True Negative (TN) -** a <u>negative</u> target feature value and <u>correct</u> prediction
- **False Positive (FP)** - a <u>positive</u> target feature value and <u>incorrect</u> prediction
- **False Negative (FN) -** a <u>negative</u> target feature value and <u>incorrect</u> prediction

$$misclasification\ rate = \frac{(FP + FN)}{(TP + TN + FP + FN)}$$

$$clasification\ accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Other measures: precision, recall, F_1,…

Confusion matrix

	Predicted		
Target		positive	negative
	positive	TP	FN
	negative	FP	TN

TPR, TNR, FPR, FNR - ?

	Predicted		
Target		Class A	Class B
	Class A	6	3
	Class B	2	9

$$true\ positive\ rate(TRP) = \frac{TP}{(TP + FN)}$$

$$true\ negative\ rate(TNR) = \frac{TN}{(TN + FP)}$$

$$false\ positive\ rate(FPR) = \frac{FP}{(TN + FP)}$$

$$false\ negative\ rate(FNR) = \frac{FN}{(TP + FN)}$$

Overall performance of a model can be captured in a single performance measure, for example, misclassification rate. To fully understand how a model is performing, it can often be useful to look beyond a single performance measure.

Performance Measures. Continuous Targets

M – model, d_i - test instances, t_i - expected target values

Mean square error (MSE)

$$MSE = \frac{\sum_{i=1}^{n}\left(t_i - M(d_i)\right)^2}{n}$$

Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(t_i - M(d_i)\right)^2}{n}}$$

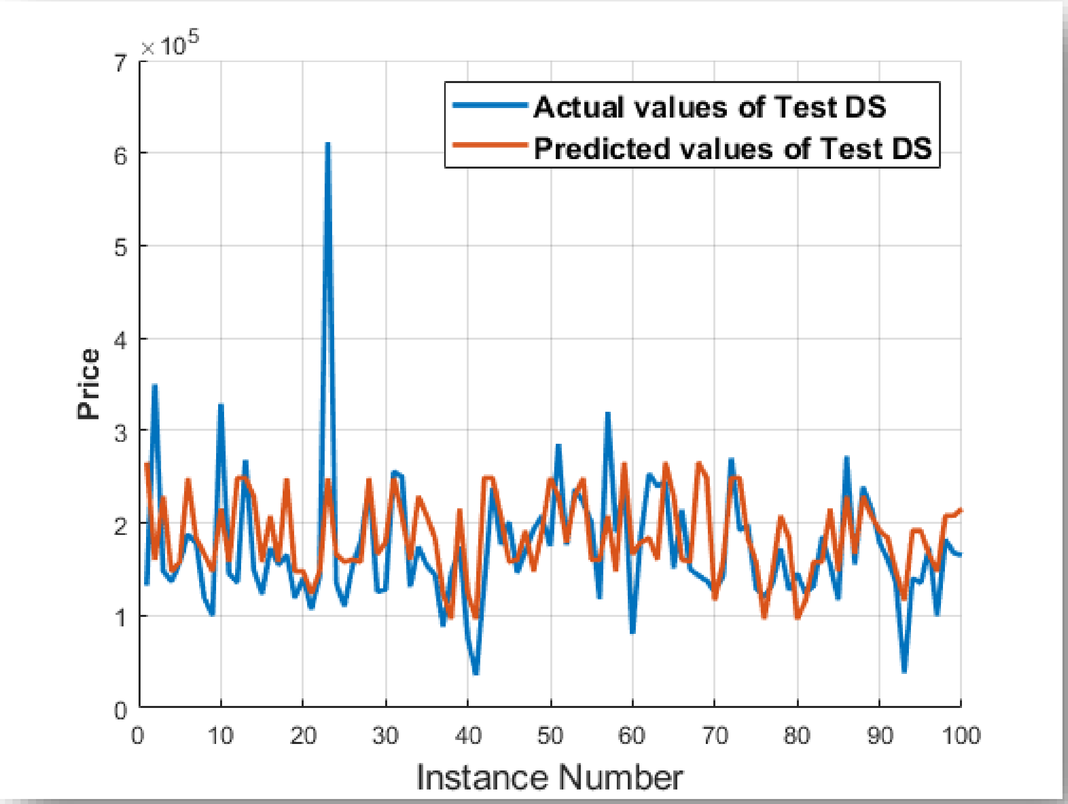
Mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{t_i - M(d_i)}{t_i}\right|$$

Other measures: mean absolute error, R^2, ...

Regression problem

Actual value representation



Object detection and Image segmentation

Actual

Predicted

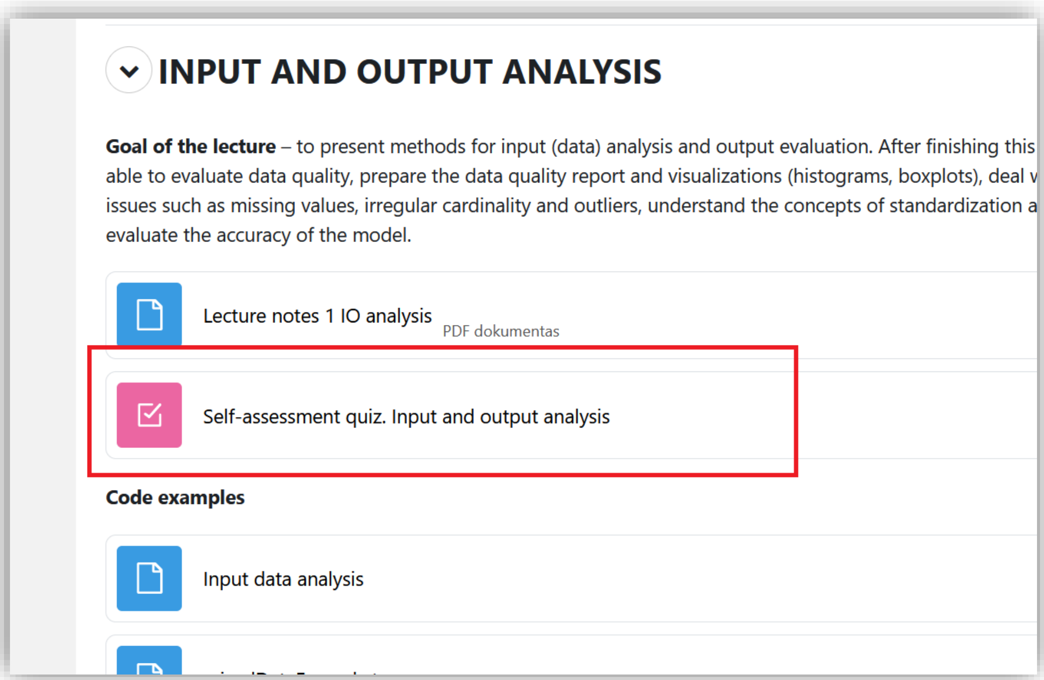
Image segmentation

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

IoU – intersection of union

Other measures: mean average precision (mAP),...

Input and output analysis



10 - 15 min.